UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

CAPACITATED LOCATION-ROUTING PROBLEMS

by

Gilbert Laporte*, Yves Nobert**
and Danielle Arpin*

* École des Hautes Études Commerciales de Montréal, 5255 avenue Decelles, Montréal, Canada H3T 1V6

** Département des Sciences administratives, Université du Québec à Montréal, 1495, rue Saint-Denis, Montréal, Canada H3C 3P8

The authors are grateful to the Canadian Natural Sciences and Engineering Research Council (grants A4747 and A5486) and to the Québec Government (F.C.A.C. grant 83EQ0428) for their financial support.

Ce rapport a été publié grâce à une subvention du fonds F.C.A.C. pour l'aide et le soutien à la recherche.

Université de Montréal

Centre de recherche sur les transports - Publication #357
Ecole des Hautes Études Commerciales - Cahier G-84-05

Mars 1984

ABSTRACT

In many location-routing problems, the objective is to locate one or many depots among n points (cities, customers) and to construct optimal (delivery) routes from depot(s) to the remaing points. The objective function is the sum of depot operating costs, vehicle costs and routing costs. This paper considers one such problem in which a non-negative weight d_i is assigned to each city i and where cities are to be visited by vehicles having a given capacity D: the solution must be such that the sum of weights of cities on any given route is not exceeded. An optimal algorithm is presented, together with numerical results.

RESUME

PROBLEMES MIXTES DE LOCALISATION ET DE CONSTRUCTION DE TOURNEES SOUS CONTRAINTES DE CAPACITE

Cet article considère un problème mixte de localisation et de tournées dans lequel des véhicules de capacité limitée doivent livrer des marchandises à un ensemble de villes. Le problème consiste à choisir parmi un ensemble de sites candidats un certain nombre de dépôts pour les camions et à construire les routes de livraison. La solution optimale est celle qui minimise la somme des coûts reliés aux dépôts, aux camions et à la distance totale parcourue. On formule le problème comme un programme linéaire en nombres entiers. Le modèle est résolu par une méthode de relaxation de contraintes insérée dans un algorithme de branch and bound. On présente des résultats numériques.

1. Introduction

The design of distribution systems frequently involves selecting sites for a number of depots (warehouses) and establishing delivery routes from the depots to sets of cities or customers. In such contexts, location and routing decisions are closely interrelated and cannot be made separately without running the risk of suboptimizing.

Location-routing problems are frequently encountered in practice. Several interesting applications are described in the surveys by Madsen [15], Perl [18] and Laporte and Nobert [11]. In contrast, solution methods for these problems are at an early stage of development when compared with the number of available algorithms for pure location or pure routing problems (see [4,5,6] and [1] for surveys of these two types of problems). This is due in large part to the difficulty of these problems, pointed out by some researchers ([19,p.248], [4,p.189]). Most algorithms for location-routing problems are approximate. The heuristic which is most often used (see [18,20,21]) consists of initially allocating customers to depots and of then constructing delivery routes. These location and routing decisions are then modified if this leads to an overall cost reduction. The process ends when no marginal improvement can be achieved.

The authors are aware of only two studies reporting the development of exact algorithms for location-routing problems. In [9], Laporte and Nobert consider the problem of simultaneously locating one depot among n nodes and of establishing m delivery routes from the depot to the remaining n-1 nodes. The problem is formulated as an integer linear program (ILP) which is solved by a constraint relaxation method; integrality is obtained by branch and bound. The largest problem solved contains 50 cities. In a more recent paper [14], Laporte et al. study more general location-routing situations involving simultaneous choice of several depots among n cities and the optimal routing of vehicles through the remaining cities. As in [9], the problems are formulated

as ILP's which are handled via constraint relaxation. This time however, integrality is reached through the gradual introduction of Gomory cutting planes (see [3] and the Land-Powell MIF program [7]). In a first series of problems, each city may be visited only once and no route may connect two depots. The difficulty of the problem is shown to be strongly related to the nature of the distance matrix and to other factors such as the absence or presence of fixed costs on the depots and the existence of an upper bound on the number of depots which may be used. In a second series of problems, multiple visits to the same city are allowed but no fixed costs or upper bounds on the number of depots are imposed. The authors show that this problem is relatively easy to solve and constitutes in fact a relaxation of the problem of finding the shortest complete cycle in a graph, previously analyzed by Miliotis et al. [16].

This paper considers a situation which is at the same time more general and more realistic than those studied in [9] and in [14]:

(i) the set of potential depots is disjoint from the set of customers;

(ii) multiple passages through the same city are allowed if they result in distance savings;

(iii) a non-negative weight may be assigned to each customer: the total weight of each route may never exceed vehicle capacity.

It is necessary at this stage to introduce some notation. Let $G = (N,E,C)$ be a graph where $N = \{1, \ldots,n\}$ is a set of nodes (representing cities or customers), where E is a set of undirected edges (i,j) and where $C = (c_{ij})$ is the symmetrical matrix of shortest distances associated with the edges. C therefore always satisfies the triangle inequality, i.e.

(1) $c_{ij} \leq c_{ik} + c_{kj}$ for all $i,j,k \in N$

Note that (i,j) and c_{ij} are only defined if $i < j$. Let $R \subseteq N$ be a set of *potential depots*. The number P of such nodes used as depots in the optimal solution must lie between two prespecified bounds $\underline{P} \geq 1$ and $\bar{P} \leq |R|$. The cost of using node r as a depot is equal to g_r. There are m_r identical vehicles based at depot r, each with the same capacity D and a fixed cost f_r. To each node i of N-R, we associate a nonnegative *requirement* d_i $(\leq D)$.

The problem consists of selecting depot sites (when $\underline{P} < |R|$), of determining how many vehicles are based at each selected depot and of establishing vehicle routes in such a way that

(i) each route starts and ends at the same depot;

(ii) all requirements are met exactly once by a vehicle (the same city i may be visited more than once if this saves distance but then, the requirement d_i of that city is satisfied only once by one vehicle);

(iii) the sum of all requirements satisfied by any vehicle does not exceed D;

(iv) $\underline{P} \leq P \leq \bar{P}$;

(v) for each node r used as a depot, the number of vehicles lies between two prespecified bounds $\underline{m}_r$ and $\bar{m}_r$: $1 \leq \underline{m}_r \leq m_r \leq \bar{m}_r$;

(vi) the total cost is minimized.

This problem is a generalization of many well known problems studied in the operational research literature. When $P = |R|$ and according to the value of $|R|$, it reduces to the travelling salesman problem [2], to the multiple travelling salesman problem [8], to the capacitated vehicle routing problems [10] or to a family of multidepot vehicle routing problems recently treated by the authors [12].

Our purpose is to provide an ILP formulation and an exact algorithm for this problem.

2. Formulation

In addition to the notation already introduced, we define

L : an arbitrarily large number

$$\lceil t \rceil = \begin{cases} \text{the smallest integer greater than or equal to } t & \text{if } t > 0 \\ 1 & \text{otherwise} \end{cases}$$

x_{ij} : a variable indicating the number of times edge (i,j) is used in the optimal solution. x_{ij} is not defined if $i \geq j$, if $i,j \in R$ or if $d_i + d_j > D$. x_{ij} must be interpreted as x_{ji} whenever $i > j$.

y_r : a binary variable indicating whether node r is used as a depot ($y_r = 1$) or not ($y_r = 0$).

The problem can be formulated as follows:

$$(P) \quad \text{minimize } z = \sum_{i,j \in N} c_{ij} x_{ij} + \sum_{r \in R} (g_r y_r + f_r m_r)$$

subject to

$$(2) \quad \sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2 \qquad (k \in N-R)$$

$$(3) \quad \sum_{i<r} x_{ir} + \sum_{r<j} x_{rj} = 2m_r \qquad (r \in R)$$

$$(4) \quad \sum_{i,j \in S} x_{ij} \leq |S| - \left\lceil \frac{\sum_{k \in S} d_k}{D} \right\rceil \qquad (S \subseteq N-R,\ |S| \geq 3)$$

$$(5)\quad x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leq 4 \qquad (i_1, i_4 \in R;\ i_2, i_3 \in N-R)$$

$$(6)\quad x_{i_1 i_2} + x_{i_{h-1} i_h} + 2 \sum_{i,j \in \{i_2, \ldots, i_{h-1}\}} x_{ij} \leq 2h-5$$
$$(h \geq 5;\ i_1, i_h \in R;\ i_2, \ldots, i_{h-1} \in N-R;\ d_{i_2} + \ldots + d_{i_{h-1}} \leq D)$$

$$(7)\quad y_r \leq m_r \leq L y_r \qquad (r \in R)$$

$$(8)\quad \underline{m}_r \leq m_r \leq \bar{m}_r \qquad (r \in R)$$

$$(9)\quad \underline{P} \leq \sum_{r \in R} y_r \leq \bar{P}$$

$$(10)\quad y_r = 0, 1 \qquad (r \in R)$$

$$(11)\quad x_{ij} = 0, 1 \qquad (i, j \in N-R)$$

$$x_{ij} = 0, 1, 2 \qquad (i \text{ or } j \in R)$$

In this formulation, constraints (2) specify that each city not used as a depot must be *serviced* exactly once by a vehicle. Of course, any city may be *visited* more than once if this saves distance, but this need not appear explicitly in the formulation. For example, if city k lies on two shortest paths (between i_1 and j_1 and between i_2 and j_2, say), then it may be visited twice, but only once will a service be provided. Similarly, constraints (3) express the fact that m_r vehicles must leave and enter each city in R. Constraints (4) ensure that the solution does not contain any illegal subtours, i.e. subtours disjoint from R or subtours having a total

weight exceeding D. These constraints generalize the *subtour elimination constraints* used by Dantzig et al. [2] for the TSP; their derivation is provided in [10] and in [17,p.48]. Constraints (5) and (6) are *chain barring constraints*. They ensure that each route starts and ends at the same depot. As their development is relatively lengthy, it is explained in the next section. The significance of the remaining constraints is fairly obvious.

3. Chain barring constraints

Consider a relaxed problem (P') obtained from (P) by removing constraints (5) and (6). The optimal solution to (P') may contain *chains* between two depots, i.e. sequences of nodes $(i_1,\ldots,i_h)$ where

(i) $i_1, i_h \in R$; $i_2,\ldots i_{h-1} \in N-R$;

(ii) $x_{i_t i_{t+1}} = 1 \qquad (t = 1,\ldots,h-1)$.

First assume that $h \geq 4$. (The two cases $h = 2$ and $h = 3$ are treated at the end of this section.) We wish to generate constraints that will eliminate such chains without cutting off any feasible solution to (P).

We describe a method for deriving such constraints. Consider a chain $H = (i_1,i_2,\ldots,i_h)$. For $\ell > k$, we define $S_{k\ell} = \{i_k, i_{k+1},\ldots,i_\ell\}$. Let $X = x_{i_1 i_2}$, $Y = x_{i_{h-1} i_h}$ and $Z = \sum_{i,j \in S_{2,h-1}} x_{ij}$ and let $\bar{X}$, $\bar{Y}$, $\bar{Z}$ be the values taken by X, Y, Z respectively. In a feasible solution, $\bar{X}, \bar{Y} \in \{0, 1, 2\}$ and $\bar{Z}$ is an integer in $[0, u(\bar{X},\bar{Y})]$ where $u(\bar{X},\bar{Y})$ an upper bound on Z whose value depends on $\bar{X}$ and $\bar{Y}$. In the following development, it is assumed that the $u(\bar{X},\bar{Y})$'s are known constants. Later in this section, we indicate how to determine the smallest bound for given $\bar{X}$ and $\bar{Y}$. Let

(12) $Q = \{(\bar{X},\bar{Y},\bar{Z}) : \bar{X},\bar{Y} \in \{0,1,2\}, \bar{Z} \in [0,u(\bar{X},\bar{Y})], \bar{Z} \text{ integer}\}$.

A valid cut is a constraint of the form

(13) $aX + bY + Z \leq d$

(where $a,b \in R$ and $d \geq 0$), which satisfies

(14) $d - a\bar{X} - b\bar{Y} \geq u(\bar{X},\bar{Y})$ $\qquad (\bar{X},\bar{Y} \in \{0,1,2\})$

while cutting the current solution characterized by $\bar{X} = 1$, $\bar{Y} = 1$ and $\bar{Z} = h-3$.

We seek the constraint which provides the deepest cut, i.e. the constraint that minimizes $(d - a - b)$, the value taken by the left hand side of (14) when $\bar{X} = \bar{Y} = 1$. The coefficients of the cut are given by the solution of (P*):

(P*) minimize $d - a - b$

subject to

(15) $d - a\bar{X} - b\bar{Y} \geq u(\bar{X},\bar{Y})$ $\qquad (\bar{X},\bar{Y} \in \{0,1,2\})$

(16) a, b free, $d \geq 0$

The optimal values of a, b and d can be obtained by means of the simplex method or by using a physical model. When (P*) possesses an infinite number of optimal solutions, we only retain the constraints which define facets of Q. Note that the cut is only effective if it eliminates the current infeasible solution, i.e. if

(17) $d - a - b < h-3$

It is easy to verify (constraints (5) and (6)) that this condition is always satisfied. We now proceed to the generation of chain barring constraints in some particular cases.

Chains involving 4 nodes

Consider a chain $H = (i_1, i_2, i_3, i_4)$. Here $X = x_{i_1 i_2}$, $Y = x_{i_3 i_4}$ and $Z = x_{i_2 i_3}$. We first observe that in any feasible solution to (P), Z can only take a positive value if $X + Y \leq 1$. Otherwise, edge (i_2, i_3) cannot enter the solution. The values of $u(\bar{X},\bar{Y})$ are therefore

$$(18)\quad u(\bar{X},\bar{Y}) = \begin{cases} 1 & \text{if } \bar{X} + \bar{Y} \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

It is easy to verify that the optimal cutting plane (13) is uniquely determined by the following three points of Q: (0, 1, 1), (1, 0, 1) and (2, 2, 0). This plane can be expressed in terms of the x_{ij}'s as

$$(19)\quad x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leq 4$$

i.e. constraint (5).

Chains involving at least 5 nodes

Now consider a chain $H = (i_1, i_2, \ldots, i_h)$ where $h \geq 5$. As above, $X = x_{i_1 i_2}$, $Y = x_{i_{h-1} i_h}$ and $Z = \sum_{i,j \in S_{2,h-1}} x_{ij}$. Since constraints (4) are satisfied, it is feasible to use only one vehicle to service all nodes of $S_{2,h-1}$ in the optimal solution (otherwise the current values of X and Y would exceed 1).

We will first illustrate the computation of $u(\bar{X},\bar{Y})$ by taking a special case. Let $\bar{X} = 2$ and $\bar{Y} \neq 2$; any feasible solution for (P) must satisfy

$$(20) \quad \sum_{j \in S_{3,h-1}} x_{i_2 j} = 0$$

and therefore

$$(21) \quad Z = \sum_{i,j \in S_{2,h-1}} x_{ij} = \sum_{i,j \in S_{3,h-1}} x_{ij} \leq (h-4) = u(\bar{X},\bar{Y})$$

By applying the same kind of reasoning, it is possible to obtain bounds $u(\bar{X},\bar{Y})$ on Z for all combinations of $\bar{X}$ and $\bar{Y}$:

$$(22) \quad u(\bar{X},\bar{Y}) = \begin{cases} h-5 & \text{if } \bar{X}+\bar{Y} = 4 \\ h-4 & \text{if } 2 \leq \bar{X}+\bar{Y} \leq 3 \\ h-3 & \text{if } \bar{X}+\bar{Y} \leq 1 \end{cases}$$

In order to find a chain barring constraint, it suffices to substitute in (P*) $\bar{X},\bar{Y}$ and $u(\bar{X},\bar{Y})$ by their values given by (22), to make the change of variable $d' = d-h+5$ and to solve the program.

The resulting constraint is given by

$$(23) \quad X + Y + 2Z \prec 2h - 5$$

i.e. constraint (6).

The two constraints described by (19) and (23) constitute facets of Q and provide the deepest cut.

Chains involving less than 4 nodes

The only chains left out of the above analysis are those involving 2 or 3 nodes. Illegal chains containing only 2 nodes cannot occur since x_{ij} is not defined when i and j $\in$ R. There is only one type of solution which could theoretically contain chains involving 3 nodes after all other types of chains have been eliminated: these would occur in subtours made up of an alternance of nodes from R and of nodes from N-R. Let $(i_1, j_1, i_2, j_2, \ldots, i_\ell, j_\ell, i_1)$ be such a subtour where $i_1, i_2, \ldots, i_\ell \in R$ and $j_1, j_2, \ldots, j_\ell \in N\text{-}R$. However, we need not consider this case since it is never disadvantageous to replace this subtour by ℓ subtours of the form (i_k^*, i_k, i_k^*) where

$$(24)\quad i_k^* = \begin{cases} j_k & \text{if } c_{i_k j_k} \leq c_{i_k j_{k-1}} \qquad (j_0 = j_\ell) \\ j_{k-1} & \text{otherwise} \end{cases}$$

since

$$(25)\quad 2c_{i_k i_k^*} < c_{i_k j_{k-1}} + c_{i_k j_k}$$

4. Algorithm

The algorithm developed to solve (P) can be summarized as follows.

Step 0: Obtain a first feasible solution by means of an appropriate heuristic. Let z* be the cost of that solution.

Step 1: Select a subproblem from the list. (The first subproblem will include constraints (2), (3), (7), (8) and (9) of (P) as well as the upper bounds on the variables.)

Step 2: Solve the subproblem using simplex (here we use the Land-Powell code [7]). Let $\bar{z}$ be the cost of the optimal solution.

Step 3: If $\bar{z} \geq z^*$, proceed to step 9.

Step 4: The current solution contains

(i) sets of nodes $\{i_1, \ldots, i_\ell\}$ $(\ell > 1)$ corresponding to chains $(i_1, \ldots, i_\ell)$ such that $\{i_2, \ldots, i_{\ell-1}\} \cap R = \phi$ if $\ell > 2$ and for which *all* variables $x_{i_1 i_2}, x_{i_2 i_3}, \ldots, x_{i_{\ell-1} i_\ell}$ have been previously *fixed* at 1, and

(ii) nodes of N-R not belonging to such chains (we define for each such node i a set $\{i\}$).

For convenience, we refer to these sets of nodes S_k (corresponding to chains or single nodes) as *components*. Each S_k has an associated weight $w(S_k)$ defined as

$$(26) \quad w(S_k) = \sum_{i \in S_k} d_i$$

Consider a component S_k. If it corresponds to a chain, let p_k and q_k be the end nodes of that chain; if it corresponds to a node i, let $p_k = q_k = i$. In the first case, $x_{p_k q_k}$ can be *forced* to zero as long as $p_k, q_k \in$ N-R. Now consider two components S_r and S_s and let $i \in \{p_r, q_r\}$, $j \in \{p_s, q_s\}$. Then x_{ij} can be forced to zero if $i, j \in$ N-R and $w(S_r) + w(S_s) > D$.

If step 4 has resulted in forcing any variable to zero, proceed to step 2.

Step 5: Check whether the current solution contains illegal subtours (i.e. subtours disconnected from R or having a total weight exceeding D). If there are no illegal subtours, proceed to step 6. Otherwise, generate a type (4) constraint for each illegal subtour and proceed to step 2.

Step 6: Check whether the current solution contains illegal chains between depots (in this context, a depot is a node r of R for which $y_r = 1$). If there are no illegal chains, proceed to step 7. Otherwise, generate for each chain a subtour elimination constraint (4) or a chain barring constraint ((5) or (6)) and proceed to step 2.

Step 7: If the solution is integer, store it and set $z^* = \bar{z}$. Proceed to step 9. Otherwise, execute the next step.

Step 8: Select a fractional variable to branch upon and create new branches in the search tree according to the procedure described in [7]. Go to step 10.

Step 9: Back up in the search tree. This consists of modifying the level of the search tree (from λ' to λ where $\lambda' \geq \lambda$) by *freeing* all variables either forced to zero or fixed at some integer value at levels $\lambda, \lambda+1, \ldots, \lambda'$. Here we used the BACKUP subroutine of the Land-Powell code [7].

Step 10: Update the list of subproblems. If the list is empty, terminate and print the best solution. Otherwise, proceed to step 1.

5. Computational results

The algorithm was tested on a number of randomly generated problems for various parameter choices. In all problems, the c_{ij}'s were defined as the straight line distance between points (X_i, Y_i) and

(X_j, Y_j) generated according to a uniform distribution on $[0,100]^2$. We then carried out three series of tests.

Tests 1: Problems involving no capacity restrictions, no vehicle costs and no fixed costs on the depots. In these problems, constraints (8) were removed and $\underline{P}$ was set equal to 1 (see table 1).

Tests 2: Weights d_i were generated according to a uniform distribution on $[0,100]$. As in previous studies [10,13], the vehicle capacity D was defined as

$$(27) \quad D = (1 - \alpha) \max_{i \in N-R} \{d_i\} + \alpha \sum_{i \in N-R} d_i$$

when α is a parameter chosen in $[0,1]$.

The remaining parameters were selected as in tests 1. Note that fixing α at 1 is equivalent to removing the capacity restriction. Setting α at a low value results in tight and relatively hard to solve problems (see table 2).

Tests 3: Problems were generated as in tests 2 but then, the effect of fixed costs on vehicles and on depots was studied. These two types of fixed costs were never used simultaneously (see table 3).

In each case, three problems were attempted for various combinations of n (number of cities), $|R|$ (number of potential depot locations) and $\bar{P}$ (maximum number of depots in the solution).

All problems were solved on the University of Montreal CYBER 855 computer, using an FTN5 compiler. Problems "failed" when they could not be solved within 300 CPU seconds. Tables 1, 2 and 3 report the results obtained.

These results indicate that problems containing up to 20-25 cities can be solved exactly by our algorithm. To the authors' knowledge,

this is the first reported attempt in the operational research literature to provide optimal solution to problems of such a complexity and possessing as many characteristics (simultaneous location and routing, capacity constraints, fixed costs on vehicles or on depots, bounds on the number of depots and on the number of vehicles per depot). The main factor explaining the success of the algorithm lies in the initial relaxation of most of the problem constraints. It can be observed that the maximum number of effective constraints in the course of the algorithm generally lies between 2n and 3n while the number of potential constraints is of the order of 2^n.

As observed in other studies [10,13], the difficulty of the problem is inversely related to the size of α. Imposing large depot costs (see table 3) also tends to produce easier problems but the same cannot be said about vehicle costs which do not seem to affect computation times one way or the other.

6. Conclusion

We have provided an integer linear programming formulation and an exact algorithm for an important class of capacitated location-routing problems. The formulation incorporates, as for many problems of the same family, degree constraints, generalized subtour elimination constraints (see [10]) and chain barring constraints.

Our results show that problems involving up to about 20-25 nodes can be solved optimally within a reasonable time. This appears to be the first time such complex problems have been tackled by means of an exact algorithm.

TABLE 1

Problems without capacity restrictions or fixed costs

n	R	S	problem number	time (seconds)	number of simplex iterations	maximum number of effective constraints	number of subtour eliminations constraints	number of chain barring constraints
10	4	3	1	14.13	799	24	11	0
			2	2.02	112	22	10	0
			3	0.97	66	20	1	0
15	6	4	1	2.50	91	29	1	0
			2	35.72	1050	34	11	0
			3	8.35	239	33	6	0
20	8	5	1	44.43	827	41	5	0
			2	22.47	368	44	9	4
			3	52.55	1056	45	12	0
25	10	6	1	> 300	2974	52	13	5
			2	> 300	2828	52	13	5
			3	38.54	492	48	3	7

TABLE 2

Problems with capacity restrictions and without fixed costs

n	|R|	p	α	problem number	time (seconds)	number of simplex iterations	maximum number of effective constraints	number of subtour eliminations constraints	number of chain barring constraints
10	4	3	0.00	1	17.75	837	28	30	14
				2	1.05	72	20	6	0
				3	24.66	1025	26	20	16
			0.25	1	57.94	2044	30	22	12
				2	> 300	10349	31	44	25
				3	5.97	326	26	12	4
			0.50	1	7.84	450	24	14	0
				2	134.48	4808	27	16	43
				3	14.61	710	25	8	16
			0.75	1	9.92	552	23	12	0
				2	19.71	1082	24	9	18
				3	1.66	102	22	5	0
			1.00	1	9.73	552	23	12	0
				2	4.51	318	22	4	0
				3	1.85	104	22	5	0
15	6	4	0.00	1	> 300	6231	44	56	20
				2	104.27	2146	44	46	45
				3	> 300	6552	50	40	41
			0.25	1	> 300	5338	39	34	17
				2	> 300	5219	38	33	12
				3	150.91	3072	34	25	12
			0.50	1	51.43	1139	34	23	7
				2	12.85	331	37	16	7
				3	11.70	353	30	7	7
			0.75	1	129.82	2570	37	19	30
				2	3.98	119	30	10	2
				3	2.56	92	26	3	3
			1.00	1	3.52	121	26	4	0
				2	2.11	76	27	6	0
				3	2.57	92	26	3	3
20	8	5	0.00	1	> 300	4163	63	72	22
				2	> 300	4239	55	59	20
				3	> 300	3352	55	59	26
			0.25	1	> 300	3068	49	23	19
				2	> 300	4600	49	46	66
				3	> 300	3214	53	47	4
			0.50	1	> 300	4244	46	8	24
				2	> 300	4634	45	14	34
				3	> 300	4120	48	42	36
			0.75	1	298.40	2882	46	6	6
				2	44.13	762	37	6	7
				3	217.67	3470	44	14	6
			1.00	1	297.3	2882	46	6	6
				2	43.18	762	37	6	7
				3	97.50	1817	39	5	0

TABLE 3

Problems with capacity restrictions and fixed costs

n	|R|	p	α	f_r (vehicle costs)	g_r (depot costs)	problem number	time (seconds)	number of simplex iterations	maximum number of effective constraints	number of subtour elimination constraints	number of chain barring constraints
10	4	3	0.50	0	0	1	7.73	450	24	14	0
						2	133.51	4808	27	16	43
						3	4.32	710	25	8	16
				50	0	1	5.39	292	23	14	0
						2	15.17	649	28	16	19
						3	3.56	174	24	6	5
				100	0	1	6.09	305	25	20	0
						2	22.68	959	28	24	27
						3	6.04	311	26	6	9
				200	0	1	6.06	305	25	20	0
						2	41.23	1509	29	20	24
						3	5.85	283	25	6	9
				0	50	1	6.25	383	27	16	0
						2	27.08	1140	27	25	15
						3	3.64	196	23	9	0
				0	100	1	5.02	291	28	17	0
						2	5.09	269	25	19	0
						3	4.79	258	23	9	0
				0	200	1	4.06	247	26	5	0
						2	5.36	258	26	20	0
						3	6.30	341	23	8	0
15	6	4	0.50	0	0	1	51.67	1139	34	23	7
						2	13.06	331	37	16	7
						3	11.63	353	30	7	7
				50	0	1	17.73	428	34	20	11
						2	147.80	2497	41	37	53
						3	156.37	2581	39	29	20
				100	0	1	17.62	396	35	18	9
						2	60.35	1142	41	23	17
						3	> 300	5125	37	50	63
				200	0	1	140.25	2272	36	34	29
						2	> 300	3527	41	47	40
						3	> 300	4583	44	58	63
				0	50	1	23.35	641	36	20	0
						2	25.97	610	38	24	0
						3	19.79	609	24	14	0
				0	100	1	22.15	585	37	19	0
						2	72.59	1094	40	34	0
						3	31.00	866	33	13	0
				0	200	1	20.13	537	37	19	0
						2	15.88	447	38	17	0
						3	28.88	891	35	14	0
20	8	5	0.75	0	0	1	298.40	2852	46	6	6
						2	44.45	782	37	6	7
						3	217.95	3470	44	14	6
				50	0	1	207.85	2407	49	19	10
						2	180.29	2136	46	19	14
						3	> 300	4370	44	9	15
				100	0	1	246.80	2452	50	24	20
						2	115.80	1482	47	18	20
						3	41.79	707	41	7	6
				200	0	1	> 300	3079	49	29	38
						2	212.31	2415	49	24	26
						3	29.84	411	40	8	
				0	50	1	87.08	1331	46	27	0
						2	94.49	1278	46	21	0
						3	42.13	706	42	14	0
				0	100	1	148.11	1846	48	21	0
						2	216.88	2728	43	17	0
						3	38.88	563	44	17	0
				0	200	1	136.84	1882	45	18	0
						2	76.16	984	45	24	0
						3	46.71	647	46	22	0

BIBLIOGRAPHY

[1] BODIN, L.D., GOLDEN, B.L., ASSAD, A., BALL, M.: "Routing and Scheduling of Vehicles and Crews. The State of the Art", Computers & Operations Research, vol.10, pp.69-211,1983.

[2] DANTZIG, G.B., FULKERSON, R., JOHNSON, S.M.: "Solution of a Large-Scale Travelling-Salesman Problem", Operations Research, vol.2, pp.393-410, 1954.

[3] GOMORY, R.: "An Algorithm for Integer Solutions to Linear Programs", Recent Advances in Mathematical Programming, pp.269-302, McGraw-Hill, 1963.

[4] HANDLER, Y.G., MIRCHANDANI, P.B.: "Location on Networks. Theory and Algorithms", MIT Press, Cambridge, Massachusetts, 233 pp., 1979.

[5] KRARUP, J., PRUZAN, T.M.: "Selected Families of Location Problems", Annals of Discrete Mathematics, vol.5, pp.327-387, 1979.

[6] KRARUP, J., PRUZAN, T.M.: "The Simple Plant Location Problem: Survey and Synthesis", European Journal of Operational Research, vol.12, pp.36-81, 1983.

[7] LAND, A.H., POWELL, S.: "FORTRAN Codes for Mathematical Programming: Linear, Quadratic and Discrete", Wiley, 249 pp., 1973.

[8] LAPORTE, G., NOBERT, Y.: "A Cutting Planes Algorithm for the m-Salesmen Problem", Journal of the Operational Research Society, vol.31, pp.1017-1023, 1980.

[9] LAPORTE, G., NOBERT, Y.: "An Exact Algorithm for Minimizing Routing and Operating Costs in Depot Location", European Journal of Operational Research, vol.6, pp.224-226, 1981.

[10] LAPORTE, G., NOBERT, Y.: "A Branch and Bound Algorithm for the Capacitated Vehicle Routing Problem", Operations Research Spektrum, vol.5, pp.77-85, 1983.

[11] LAPORTE, G., NOBERT, Y.: "Les problèmes mixtes de localisation et de construction de tournées: un état de la question", forthcoming, 1984.

[12] LAPORTE, G., NOBERT, Y., ARPIN, D.: "Optimal Solutions to Capacitated Multidepot Vehicle Routing Problems", Cahiers du GERAD G-84-02, Ecole des Hautes Etudes Commerciales de Montréal, 11 pp., 1984.

[13] LAPORTE, G., NOBERT, Y., DESROCHERS, M.: "Optimal Routing under Capacity and Distance Restrictions", Cahiers du GERAD G-83-18, Ecole des Hautes Etudes Commerciales de Montréal, 23 pp., 1983.

[14] LAPORTE, G., NOBERT, Y., PELLETIER, P.: "Hamiltonian Location Problems", European Journal of Operational Research, vol.12, pp.82-89, 1983.

[15] MADSEN, O.B.G.: "Methods for Solving Combined Two Level Location-Routing Problems of Realistic Dimensions", European Journal of Operational Research, vol.12, pp.295-301, 1983.

[16] MILIOTIS, P., LAPORTE, G., NOBERT, Y.: "Computational Comparison of Two Methods for Finding the Shortest Complete Cycle or Circuit in a Graph", RAIRO, vol.15, pp.233-239, 1981.

[17] NOBERT, Y.: "Construction d'algorithmes optimaux pour des extensions au problème du voyageur de commerce", Thèse de doctorat, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 242 pp., 1982.

[18] PERL, J.: "A Unified Warehouse Location-Routing Problem", Ph.D. Dissertation, Department of Civil Engineering, Northwestern University, Evanston, Illinois, 1983.

[19] RAND, G.K.: "Methodological Choices in Depot Location Studies", Operational Research Quarterly, vol.27, pp.241-249, 1976.

[20] TDF (Transportforskingsdelegationen): "Distribution Planning Using Mathematical Methods", Report 1977: 1, Contract Research Group for Applied Mathematics, Royal Institute of Technology, Sweden, 1977.

[21] WATSON-GANDY, C.D.T., DORN, P.J.: "Depot Location with Van Salesmen. A Practical Approach", Omega, vol.1, pp.321-329, 1973.